

freebsd-arch: Re: [Fwd: What do people think about not installing a stripped /kernel ?]

## Re: [Fwd: What do people think about not installing a stripped /kernel ?]

**Source:** <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/arch/2004-10/0150.html>

---

**From:** Maxim Sobolev (*sobomax\_at\_portaone.com*)

**Date:** 10/21/04

Date: Thu, 21 Oct 2004 01:14:01 +0300

To: Scott Long <scottl@FreeBSD.ORG>

Scott Long wrote:

> *Ruslan Ermilov wrote:*

>

>> *On Wed, Oct 20, 2004 at 07:13:35PM +0200, Max Laier wrote:*

>>

>>> *Why is this discussion ongoing? The consensus seems pretty clear:*

>>> *"Implement it, but have a make.conf option to turn it off." If there*

>>> *is concern with this make if default to off and have an option to*

>>> *turn it on.*

>>>

>>

>> *Implementing this is very easy, since it's already implemented,*

>> *just not by default.*

>>

>> *What everyone seem to have forgotten is that we also have modules,*

>> *and in the "config -g" case, we also build debug versions of the*

>> *modules. And if we're also going to install modules with debug*

>> *symbols, I think this puts the requirement for the root file*

>> *system way beyond the rational limits.*

>>

>>

>> *Cheers,*

>

>

> *I tend to agree. What do you think of my proposal to have installkernel*

> *(optionally or whatever) put unstripped binaries somewhere outside of the*

> *root partition?*

OK, I've just checked objcopy manpage and found that there is actually a better way which combines best properties of both approach. In modern GNU toolchain it is possible to split executable and debugging info into two separate files, but put a reference into executable, so that you don't have to worry about how to load debugging symbols:

--only-keep-debug

Re: [Fwd: What do people think about not installing a stripped /kernel ?]

freebsd-arch: Re: [Fwd: What do people think about not installing a stripped /kernel ?]

Strip a file, removing any sections that would be stripped by  
--strip-debug and leaving the debugging sections.

The intention is that this option will be used in conjunction with  
--add-gnu-debuglink to create a two part executable. One a  
stripped binary which will occupy less space in RAM and in a dis-  
tribution and the second a debugging information file which is only  
needed if debugging abilities are required. The suggested proce-  
dure to create these files is as follows:

1.<Link the executable as normal. Assuming that is is called>  
"foo" then...

1.<Run "objcopy --only-keep-debug foo foo.dbg" to>  
create a file containing the debugging info.

1.<Run "objcopy --strip-debug foo" to create a>  
stripped executable.

1.<Run "objcopy --add-gnu-debuglink=foo.dbg foo">  
to add a link to the debugging info into the stripped exe-  
cutable.

I checked, this works like a charm with our current toolchain/gdb. This  
allows us to do the following clever trick WRT kernel debug:

1. Compile kernel/modules with debugging symbols;
2. Split out executable and debugging pieces for each module;
3. Associate each executable file with appropriate debug file;
4. Install executable into /boot/kernel as usually;
5. Install real debugging into /var/something, put symlink to it into  
/boot/kernel.

By the way, this approach can be extended to be an option of buildworld  
as well! It can be good way to trade developers' time for some hdd  
space, since with this option "on" you will always be able to debug  
misbehaving application/library without the need to recompile/reinstall  
everything!

Opinions?

-Maxim

---

freebsd-arch@freebsd.org mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd-arch>

To unsubscribe, send any mail to "freebsd-arch-unsubscribe@freebsd.org"